BWN技术手册-中文

1. 软硬件背景
   1. 深度学习在语音识别中的应用

随着以深度学习和人工神经网络为代表的人工智能技术不断的完善和发展，语音技术开始逐步使用LSTM（长短期神经网络）、RNN（循环神经网络）和CNN（卷积神经网络）来完成语音识别、声纹识别与实时翻译等技术。新模式算法极大的提高了语音识别的性能，并得以在工业界大规模应用，比较典型的应用有科大讯飞实时翻译、百度语音等。

* 1. FPGA可定制逻辑在深度学习加速中的应用

深度学习虽然在算法性能上取得了较大的突破，但也对算力提出了更高的要求。传统CPU面对并行性高、计算密集访存密集的深度学习计算效率较低；GPU处理器效率较CPU高，但功耗与价格居高不下；ASIC可以在功耗与就散性能上取得较好的平衡，但设计流程长、灵活性差、设计成本高。

对比之下，FPGA是一种可编程、可用户定制化的现代大规模集成电路产品。设计者在通过诸如Verilog HDL的硬件描述语言完成电路设计后，通过相应的软件平台进行仿真、综合与布局后生成比特流描述文件。将比特流描述文件烧写至FPGA中的Flash模块即可完成门电路级别的编程逻辑控制。

对于计算机体系结构与系统结构的研究者和设计者来说，FPGA虽然实际的工作频率远低于实际CPU的工作频率，但FPGA成本低廉使用灵活的特性可以让用户针对性的设计最有效率的加速器。

1. 算法背景
   1. 运行平台与数据集

本程序运行在Matlab 2018a平台。Matlab 2018a 包含完整的并行计算库和音频处理相关库，相比之前的版本能更好的完成语音识别的任务。本程序的数据集采用谷歌TensorFlow中的开源音频库，该数据集顺序存储了若干单词的语音片段，每个单词都分别对应若干音频文件并标记其label。本语音模型针对{‘yes’, ‘up’, ‘down’, ‘right’, ‘left’, ‘unknown’}六个单词标签进行识别。

数据集的存储模式如图1所示：



图1

* 1. 神经网络算法与程序流程

本语音识别模型的核心为一个针对语音帧特征图的卷积神经网络。程序首先从数据集中随机筛选{‘yes’, ‘up’, ‘down’, ‘right’, ‘left’, ‘unknown’}六种单词标签的音频样本，再调用Matlab 2018a语音库中的MFCC算法对语音文件进行预处理，将序列化的音频转为20\*49\*1的语音特征图矩阵。该矩阵将被送入一个由两个卷积层和三个全连接层组成的卷积神经网络中，最终通过softmax函数输出预测结果。图2中详细描述了该卷积神经网络的结构。

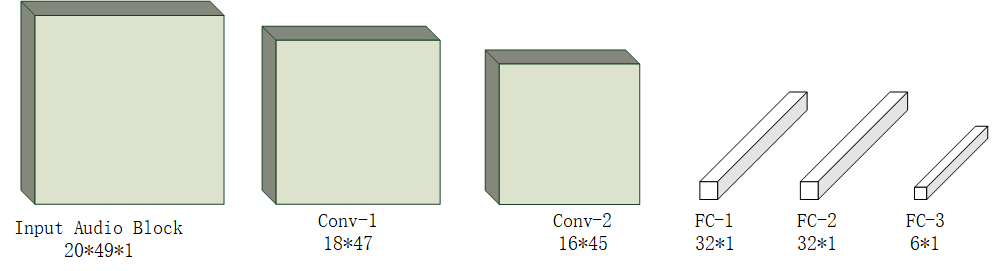


图2

神经网络的两个卷积层步长为1，卷积核为3\*3，通道数均为32，且不存在padding补偿。神经网络采用的激活函数为ReLU，且在每一层中均有批正则化操作用于改善数据分布。该模型具有网络规模小、参数规模大的特点（表1列出了该卷积神经网络各层的参数规模）。但该模型中的权值与激活值均为浮点小数，在FPGA上加速时会造成计算资源紧张、浮点计算延迟高的问题；另外在浮点数据格式下，模型参数的存储需求超出了FPGA片内存储，而将参数存于片外存储器上又会增加访存开销、降低加速性能。因此从上述两方面考虑，该模型还需要进行量化处理。



表1

* 1. 网络量化

为了解决上述问题，本设计使用两种量化方法优化算法原型：（1）权值的二值化；（2）激活值的定点化。

本设计将网络中的以浮点数格式表示的原始权值通过一系列处理（具体流程见图3）转化为表示值为+1, -1的二值数据。二值化的数据在硬件上可以通过一个二进制位的01状态来储存，即一个64位浮点权值在二值量化后只需一个比特位就可存储，大大减少了模型参数的存储需求。二值化后输出的权值形如图4.

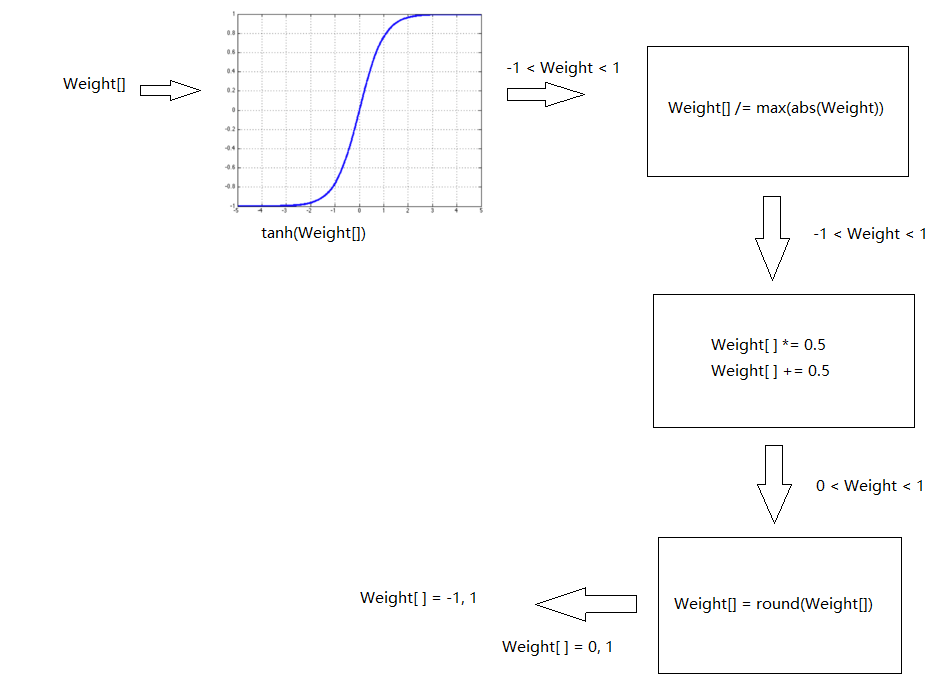


图3

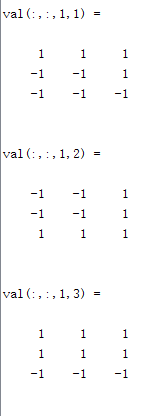
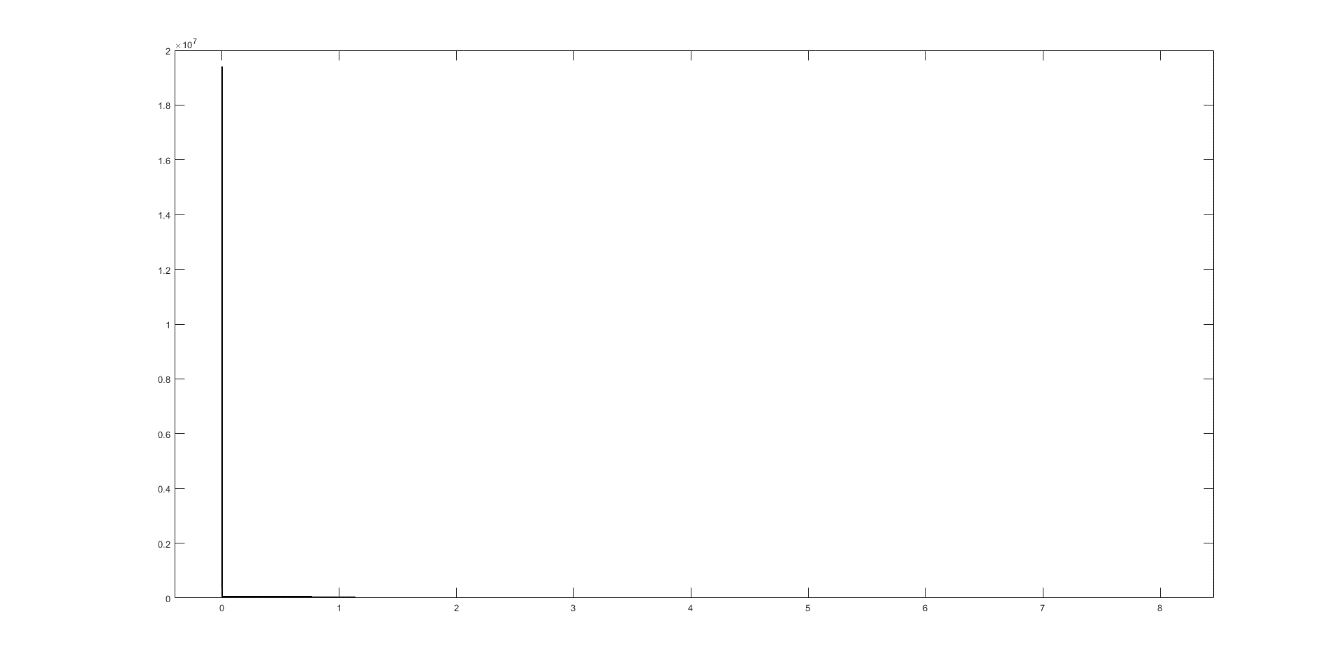


图4

在原网络中，激活值的数据格式也为64位浮点小数，在FPGA加速场景下，浮点小数不仅计算资源占用高，而且计算大，不利于设计与实现，因此本设计对激活值进行定点量化。定点量化将浮点小数转换为定点小数，通过调整小数位宽尽可能的反映真实的激活值数值、降低精度损失。

批正则化操作可以将每一层的输出数值范围收缩到一个较小的区间，通过多次试验可以确定每一层的大概输出数值区间，进而确定每一层激活值定点量化的整数部分位宽。但小数位宽的选取则更复杂，小数部分位宽增大，有利于定点量化后的数值更真实的反映原始激活结果的数值，但当小数位宽过大时则会增大数据的整体位宽，不利于规整的硬件实现；而当小数位宽减小时，则会带来较大的数值损失，影响模型的整体精度性能。

此外，批正则化操作在神经网络中起到重新分布数据的重要作用，而数据分布规律对于模型的精度表现有十分重要的影响。图5展示了定点量化（定点量化方案为小数位宽7bit）前后卷积层2输出数据的分布差异。对比后可以发现，相比原网络，量化后数据分布离散化程度更高，噪音更多。原网络的数据分布更平均、更平滑且噪音少。在该定点量化方案下， unknown标签的精度下降了5%，left和up标签的判断精度则下降的更多。批正则化操作设计乘除法与开方运算，对数据精度更敏感，因此实验中往往需要给正则化参数（均值、方差）更多的小数位宽以保证模型的整体精度。



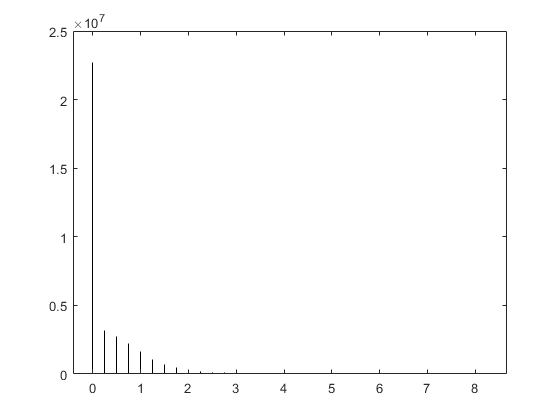


图5 正则化并RELU后的数据分布，上为原网络，下为量化后

为了探索最优的定点量化方案，本设计在固定整数位宽和单个量化数据位宽不大于32位不小于16位的限制条件下，对小数位宽方案进行逐一逐层的实验，最终得出最佳量化方案：激活值小数位宽为8/正则化参数小数位宽为9、整数部分参数依据每层的数据分布区间浮动，整体语音识别精度不小于85%。在增加量化功能后，整体程序的运行流程如图6.

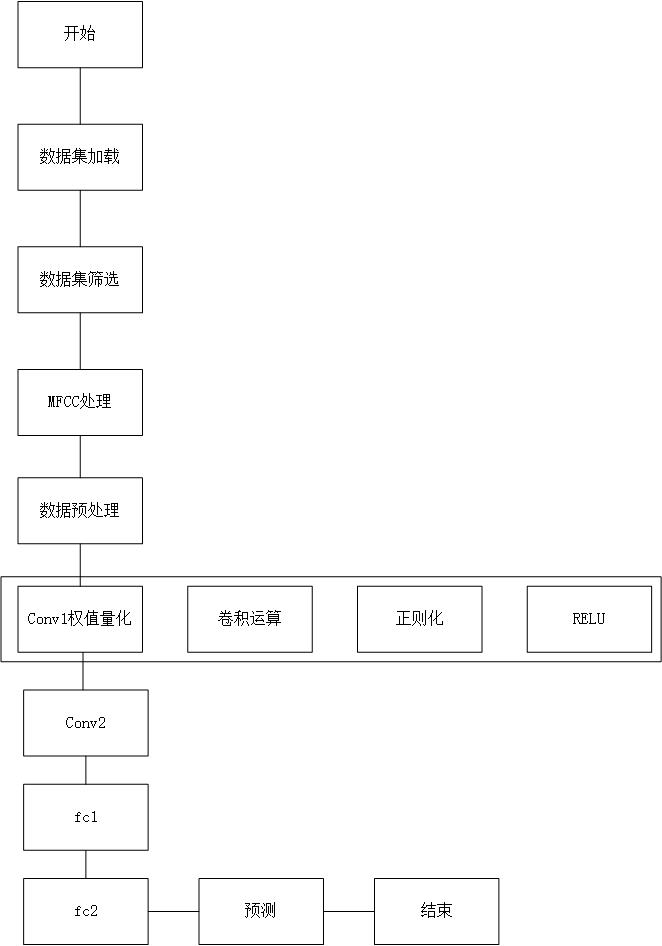


图6

* 1. 算法运行时间

我们在8700k平台上使用Matlab单线程和多线程库分别对该程序进行分段计时，结果如下表2。



表2

当程序为单核单线程时，程序每次运行平均处理1523.6个样本，全流程平均运行时间为127.1163秒，平均每个样本的处理时间为83.43毫秒。

当程序为4核并行运行时，测试结果如下图：



表2

并行加速后程序全流程平均运行时间为49.667秒，平均每个样本的处理时间为32.21毫秒。

1. 硬件总体设计

由于目标模型是小的，浅的，二值化的，所以我们采用的主要方法是通过在片上设置共享的二值化参数，逐级加速神经网络，以及在级间设计流水线来降低存储延迟。

* 1. 参数存储设计

与一些神经网络加速器需要用DRAM代替片上存储器来存储参数不同，我们的模型具有较小的参数。 表3显示了该模型参数的详细信息。



表3

FC-1层占据了大部分参数大小，而其他层的数据相当微小，可以直接存储在片上。考虑到FC-1参数的规模，在多个PE之间共享它们是很合理的设计方案。我们将所有PE设置为同步工作，并在FC-1的计算时提取完全相同的预训练数据。共享数据块由32个BRAM组成，每个BRAM用于FC-1中的一个卷积核。

BRAM中的参数沿第三维存储，与FC-1层的输入数据组织方式一致。 通过广播，共享参数数据被发送到所有PE。在图7中展示了参数存储的一些细节，PE向参数共享模块发送目标参数loaf地址，该地址是Bram块的行地址。

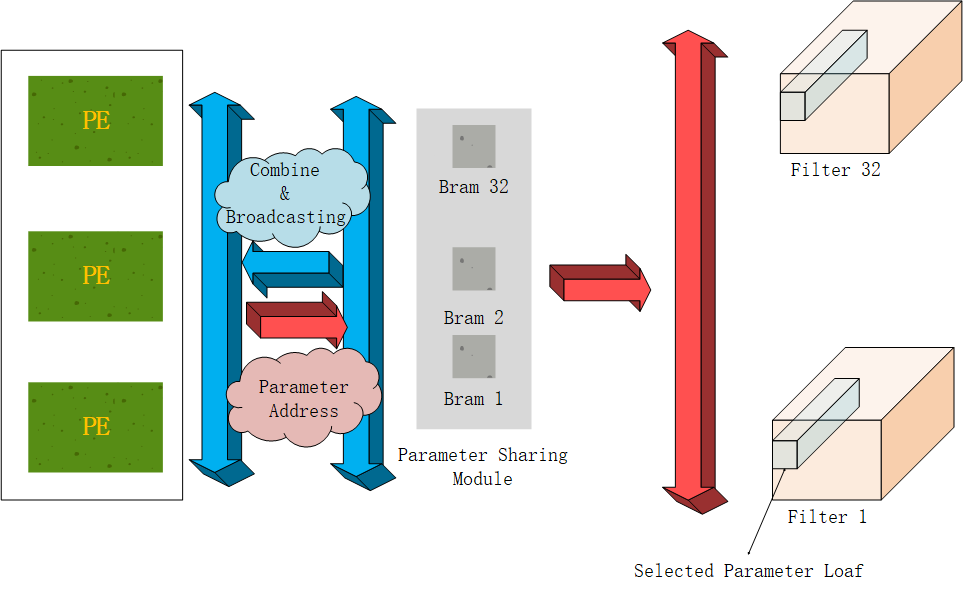


图7

* 1. 位宽扩展

在一个卷积层或全连通层内，输入的定点特征数据通常服从正态分布，且变化范围较小，而经过MAC后，数据的方差可能会很大且不规则。 批量归一化依赖于这些中间结果的均值和方差来修改MAC后的数据分布，这对最终的精度至关重要。 如果在归一化步骤中仍然使用与输入特征相同的数据格式，将会给最终结果带来不必要的损失。

为了解决这一问题，我们引入了位宽扩展，在计算时给特征数据和归一化参数增加额外的十进制数宽。 DSP输出乘法运算结果后，对多出的十进制位宽进行截断，使数据恢复到原始输入格式。

MAC处理的中间结果也有类似的位宽扩展需求。 不同之处在于，MAC的结果需要更多的整数位宽而不是十进制位宽。 对归一化步和MAC累加器进行了扩展，结果表明该方法在硬件上保证了计算精度。

* 1. 层间流水

实现逐级流水线的关键是要平衡级间的运行周期。在VGG-16和AlexNet等深度卷积神经网络中，很难保持这种平衡，因为随着网络的深入，深度层将要求前面的层以更快的速度生成特征结果，这超出了当前的计算能力限制。

目标神经网络是浅层的，规模较小的，所以相对容易在两个卷积层之间保持平衡。 改变了Conv-1层的取数据地址生成方法，使之与Conv-2层的计算模式相配合。 同时，扩展了Conv-1层功能模块的并行度，实现了在一级流水线节拍内生成一系列Conv-2层的输入数据。图8展示了平衡层间流水的设计方法。

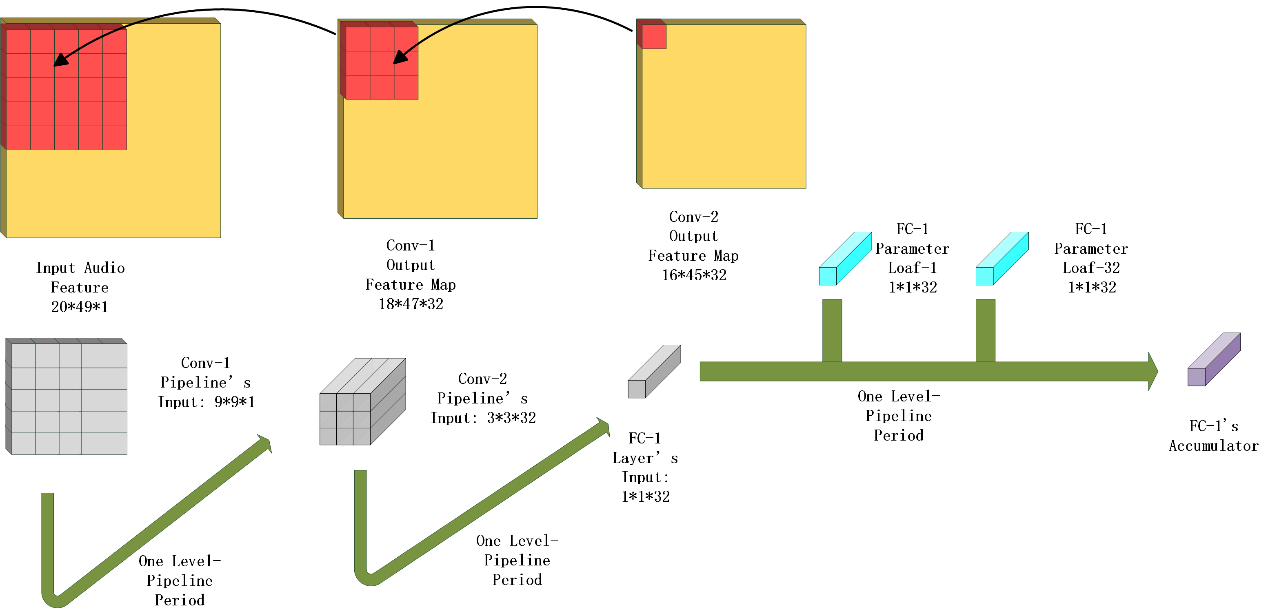


图8

为了在Conv-2层中产生一个结果，该神经网络必须在Conv-1的输出映射上计算一个3\*3滑动窗口，并从输入音频特征的5\*5区域中计算九个3\*3滑动窗口。我们扩展了Conv-1的计算阵列规模，使其在一个宏流水线周期内产生九个32\*1向量，在下一个周期内这些向量将被发送到Conv-2，并为FC-1层产生一个32\*1向量。Conv-2的功能也在一个宏周期内运行。 FC-1层是该加速器的瓶颈，由于计算量巨大，要保持该层与两个卷积层的平衡是难以承受的，但我们可以保证FC-1层在每个宏周期内保持一个向量进行累加。

通过对层次流水线的调整，目标神经网络可以在不需要将层次间结果放入DRAM中的情况下进行加速，从而降低存储开销，也就是说，从输入音频特征到最终预测结果的数据流始终保持在层次流水线中。加速器只在取数和最后回写时与DRAM通信。 在流水线的各个层次中，我们还将所有的计算分解为向量计算单元，归一化单元等功能部件，提高了硬件的运行频率。

图9展示了FPGA加速器的整体设计结构。一个加速器包含一个或多个PEs，一个DDR总线和一个DRAM（本图未显示）。每一个PE都有自己的功能模块，包括五层神经网络处理单元，地址发生器，输入输出数据存储。 Conv-1模块，Conv-2模块和FC-1模块的累加器采用流水线方式工作。

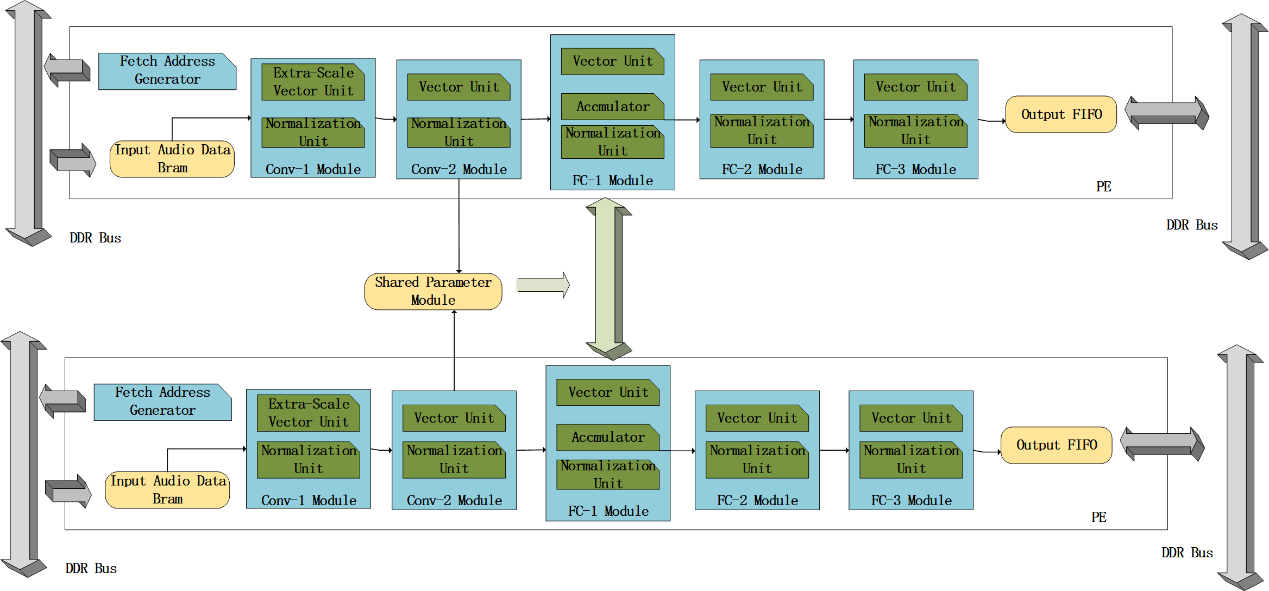


图9

1. 硬件模块设计
   1. 地址生成模块

该模块的主要作用是按照3.3中的设计方法生成对应顺序的取值地址。（此时当前音频特征图已经被预取到Input Audio Data Bram中）

主要输入：启动信号；

主要输出：地址信号->Input Audio Data Bram；

数据有效信号->Conv-1模块；

对于一个20\*49\*1的特征图矩阵，其取值地址层次可以被划分为尺寸为5\*5\*1的外层窗口、尺寸为3\*3\*1的卷积窗口和尺寸为1\*3\*1的卷积行。外层窗口和卷积窗口都按照步长为1的典型的卷积神经网络中滑动窗口的模式进行逐个更新。算法1中详细介绍了三种取值层次的更新过程。由于地址生成模块一次计算三个卷积行的地址，因此Input Audio Bram会有三个完全一样的存储块，每块都完整的存储了当前要计算的音频特征图，Input Audio Bram模块可以同时对三个卷积行地址进行取数。

当启动信号到来后，地址生成模块开始计算预取数据地址，并根据对应的取值延迟像Conv-1模块发送“数据有效”信号。Input Audio Data Bram地址后开始取数据并传送到Conv-1模块。Conv-1模块在同时收到数据和有效信号后开始计算功能。

若当前特征图未读完：

若当前外层窗口未读完：

若当前卷积窗口未读完：

当前指针位 = 1；

若当前指针位置 < 3：

根据指针位置生成第一行行地址；

根据指针位置生成第一行行地址；

根据指针位置生成第一行行地址；

当前指针位置 += 1；

当前卷积窗口按步长=1步进；

当前外层窗口按步长=1步进；

完成所有取值，地址生成模块复位；

算法1

* 1. Conv-1模块

该模块的主要功能是完成Conv-1的计算任务。该模块由Conv-1二值化参数存储、二值向量-定点向量运算器、加法树和正则化模块组成。

主要输入：启动信号；

特征图数据；

特征图数据有效信号；

主要输出：Conv-1计算结果；

Conv-1计算结果有效信号；

在Conv-1模块收到特征图有效信号后，将当前的数据结果进行缓存，直到收到3组特征图数据有效信号，得到完整的一个卷积窗口中的9个数据后，Conv-1模块才会启动后续计算。

Conv-1中共有32个二值向量-定点向量运算器，9个定点卷积窗口数据被广播到32个向量运算器中；同时二值化参数存储寄存器中存储了32个卷积过滤器的二值化参数数据，每个滤波器的参数都被传送到对应的向量运算器中与定点数据进行运算。每个向量运算器计算得到的数据经过并行加法树后输出1个16位定点数据。此时Conv-1模块得到了32个16位定点运算结果。

每次运算得到的32\*16bit\*1向量会被送入正则化模块，依次进行减去平均值、除方差的平方根、乘offset系数和加上偏移量超参数的运算，最终通过判断符号位完成ReLU功能，得到当前Conv-1的输出。该输出仍为32\*16bit\*1，并同Conv-1计算结果有效信号一起被送入Conv-2模块。图10中展示了Conv-1模块的内部组成。

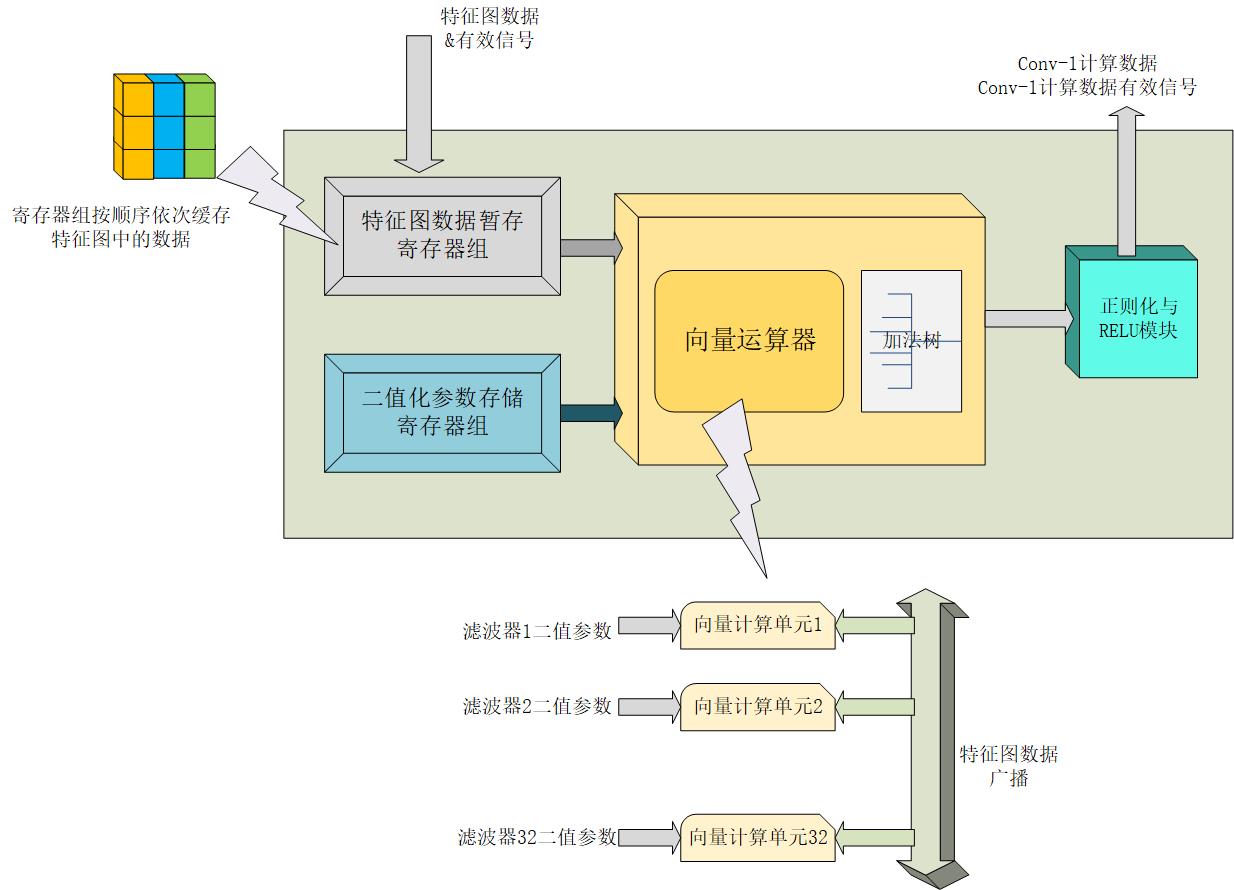


图10

* 1. Conv-2模块

该模块的主要功能是完成Conv-2的计算任务。该模块由Conv-2二值化参数存储、二值向量-定点向量运算器、加法树、累加器、共享权值地址计算模块和正则化模块组成。

主要输入：启动信号；

Conv-1计算数据；

Conv-1计算数据有效信号；

主要输出：Conv-2计算结果；

Conv-2计算结果有效信号；

共享权值地址；

共享权值地址有效信号；

在Conv-2模块收到Conv-1计算结果有效信号后，启动当前结果向量的运算。

Conv-2中共有32个二值向量-定点向量运算器，维度为32\*16bit\*1的Conv-1计算输出向量数据被广播到32个向量运算器中；同时二值化参数存储寄存器中存储了32个卷积过滤器的二值化参数数据，每个滤波器的参数沿第三个维度与依次被传入的Conv-1数据向量一一对应，32个滤波器中的权值向量被传送到对应的向量运算器中与定点数据进行运算。每个向量运算器计算得到的数据经过并行加法树后输出1个16位定点数据。此时Conv-2模块得到了32个16位定点运算结果。

第一次运算得到的32个16位定点数据向量被存入累加器中，之后每8次运算的中间计算结果都会在累加器中与累加和进行求和。

当累加器中存储了9次运算结果的累加和后，累加器清零，每次运算得到的32\*16bit\*1向量会被送入正则化模块，依次进行减去平均值、除方差的平方根、乘offset系数和加上偏移量超参数的运算，最终通过判断符号位完成ReLU功能，得到当前Conv-2的输出。该输出仍为32\*16bit\*1，并同Conv-2计算结果有效信号一起被送入FC-1模块。同时Conv-2模块会向共享权值模块发送权值索引地址。图11展示了Conv-2模块的内部组成。

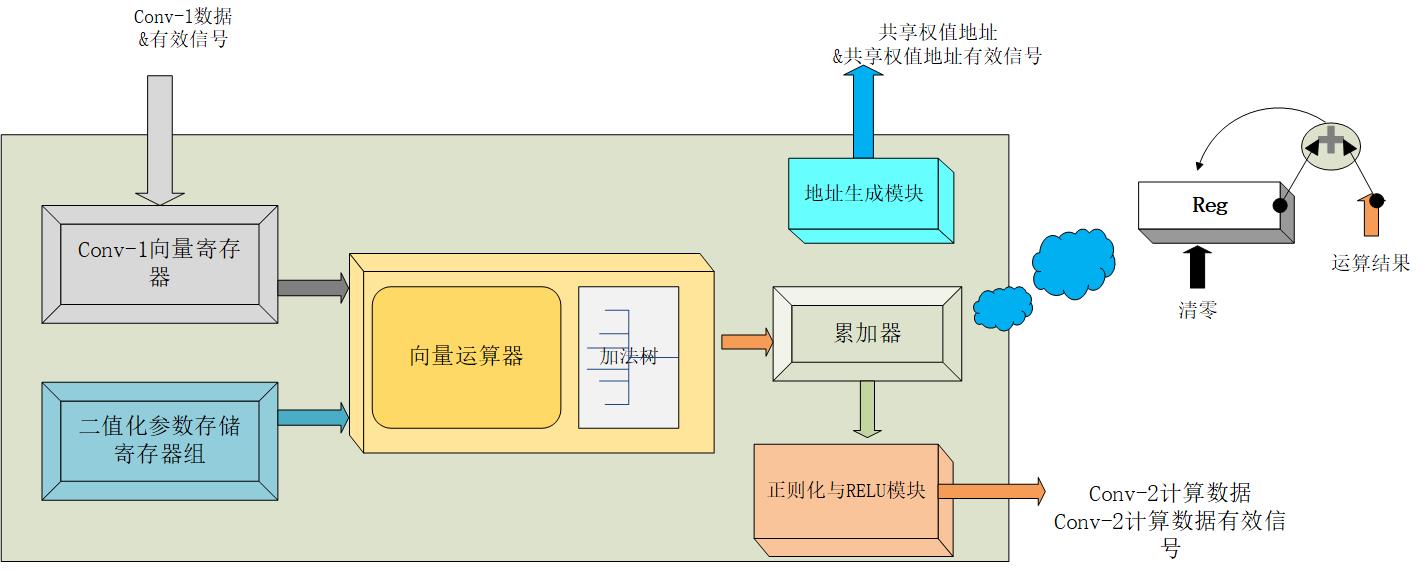


图11

* 1. 权值共享模块

权值共享模块的主要功能是存储规模较大的FC-1共享参数。

主要输入：共享参数地址；

共享参数地址有效信号；

主要输出：共享参数数据；

共享参数数据有效信号；

当共享权值模块收到共享参数地址和共享参数地址有效信号后，即按照索引地址到BRAM存储器取对应数据。当数据取出后，数据同有效信号一起发送给各个PE。共享权值模块的设计细节已在本文3.1中详述。

* 1. FC-1模块

FC-1模块的主要功能是完成数据从特征图到向量的展开计算。该模块由二值向量-定点向量运算器、加法树、累加器和正则化模块组成。

主要输入：共享权值；

共享权值有效信号；

Conv-2输出结果；

Conv-2输出结果有效信号；

主要输出：FC-1输出结果；

FC-1输出结果有效信号；

在FC-1模块收到Conv-2计算结果有效信号和共享权值有效信号后，启动当前结果向量的运算。

FC-1中共有32个二值向量-定点向量运算器，维度为32\*16bit\*1的Conv-2计算输出向量数据被广播到32个向量运算器中；同时来自共享权值模块的二值化参数中包含了32个卷积过滤器的二值化参数数据，每个滤波器的参数沿第三个维度与依次被传入的Conv-2数据向量一一对应，32个滤波器中的权值向量被传送到对应的向量运算器中与定点数据进行运算。每个向量运算器计算得到的数据经过并行加法树后输出1个16位定点数据。此时FC-2模块得到了32个16位定点运算结果。

第一次运算得到的32个16位定点数据向量被存入累加器中，之后每次运算的中间计算结果都会在累加器中与累加和进行求和，直到维度为16\*45\*32的Conv-2输出特征图被完全展开成32维向量。

累加器最终运算得到的32\*16bit\*1向量会被送入正则化模块，依次进行减去平均值、除方差的平方根、乘offset系数和加上偏移量超参数的运算，最终通过判断符号位完成ReLU功能，得到当前FC-1的输出。该输出仍为32\*16bit\*1，并同FC-1计算结果有效信号一起被送入FC-2模块。图12展示了Conv-2模块的内部组成。

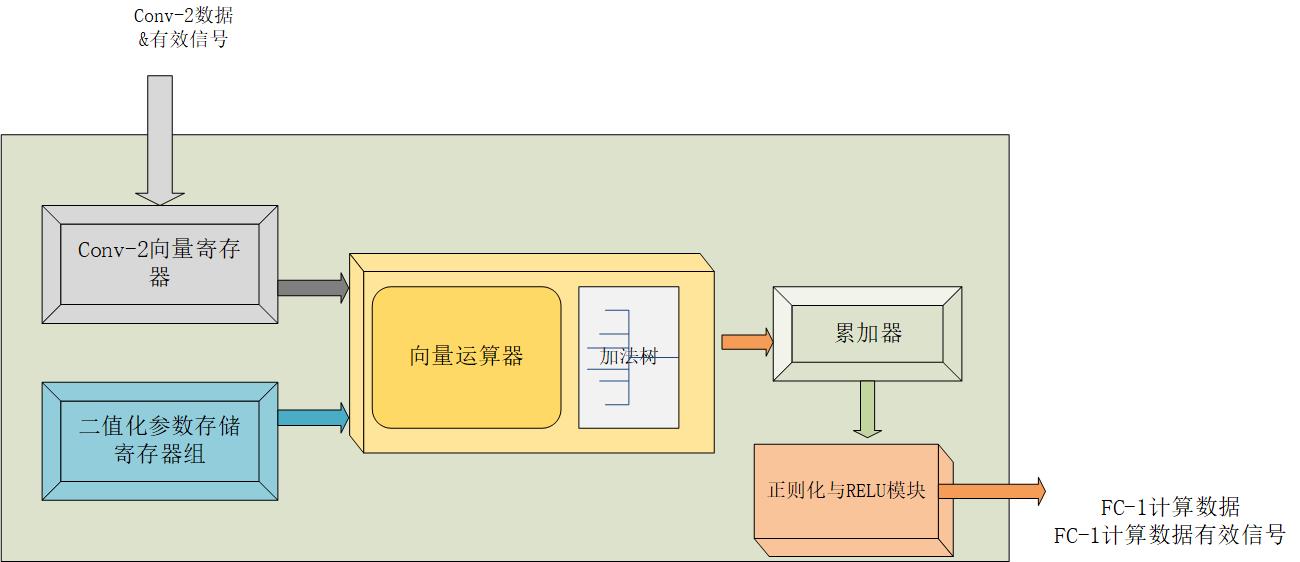


图12

* 1. FC-2模块

FC-2模块的主要功能是完成第2全连接层计算。该模块由二值向量-定点向量运算器、二值化参数存储寄存器、加法树和正则化模块组成。

主要输入：FC-1输出结果；

FC-1输出结果有效信号；

主要输出：FC-2输出结果；

FC-2输出结果有效信号；

在FC-2模块收到FC-1计算结果有效信号和共享权值有效信号后，启动当前结果向量的运算。

FC-2中共有32个二值向量-定点向量运算器，维度为32\*16bit\*1的FC-1计算输出向量数据被广播到32个向量运算器中；同时来自二值化存储寄存器组的权值被传送到对应的向量运算器中与定点数据进行运算。每个向量运算器计算得到的数据经过并行加法树后输出1个16位定点数据。此时FC-2模块得到了32个16位定点运算结果。

运算得到的32\*16bit\*1向量会被送入正则化模块，依次进行减去平均值、除方差的平方根、乘offset系数和加上偏移量超参数的运算，最终通过判断符号位完成ReLU功能，得到当前FC-2的输出。该输出仍为32\*16bit\*1，并同FC-2计算结果有效信号一起被送入FC-3模块。

* 1. FC-3模块

FC-3模块的主要功能是完成第3全连接层计算。该模块由二值向量-定点向量运算器、二值化参数存储寄存器和加法树组成。

主要输入：FC-2输出结果；

FC-2输出结果有效信号；

主要输出：FC-3输出结果；

当前特征图运算完成信号；

在FC-3模块收到FC-2计算结果有效信号和共享权值有效信号后，启动当前结果向量的运算。当该层运算完成后，将会产生PE内部的当前特征图运算完成信号。

FC-3中共有6个二值向量-定点向量运算器，维度为32\*16bit\*1的FC-2计算输出向量数据被广播到6个向量运算器中；同时来自二值化存储寄存器组的权值被传送到对应的向量运算器中与定点数据进行运算。每个向量运算器计算得到的数据经过并行加法树后输出1个16位定点数据。此时FC-2模块得到了6个16位定点运算结果并作为当前PE的最终输出交付FIFO。

1. 加速器性能指标

|  |  |
| --- | --- |
| Running Time (Seconds) | 0.4116 |
| Static Power (W) | 9.63 |
| Dynamic Power (W) | 10.06 |
| Accuracy | 84.96% |
| Utilization | 38% |
| Accelerating Ratio (8700K) | 308.83 |
| Accelerating Ratio (8700K & Parallelsim) | 120.67 |
| Throughput (Audio Segment Per Sec) | 3675.9 |
| Running Frequency (MHz) | 150 |